This application is submitted in the name of the following inventor(s):

Inventor	*Citizenship*	*Residence City and State*
Andreas V. BECHTOLSHEIM	Germany	Stanford, California
David R. CHERITON	Canada	Palo Alto, California

The assignee is *Cisco Technology, Inc.*, a California corporation having an office at 170 West Tasman Drive, San Jose CA 95134.

Title of the Invention

Packet Processing Engine Architecture

Background of the Invention

1. Field of the Invention

This invention relates to packet processing.

2. *Related Art*

In a computer network for transmitting information, messages are received by each router (or switch) at an input interface and retransmitted at an output interface, so as to forward those messages onward to their respective destinations. Each router performs a lookup operation for each message it encounters, in which the router determines from the message to which output interface the message should be forwarded.

One problem in the known art is that the lookup operation can be relatively complex, and can use a relatively large amount of processor resources. For example, the lookup operation can be complicated by concurrently determining one or more of the following:

- which output interface is the closest, within a defined network topology, to the specified destination;
- whether the message is unicast or multicast, and in the latter case, from which input interface the message was received;
- whether the message is authorized to be forwarded by this router from its specified source, and whether the message is authorized to be forwarded by this router to its specified destination;

- whether the message should be forwarded to a selected output interface for quality of service considerations, security considerations, or other administrative considerations;

- whether the message should be counted, measured, or otherwise accounted for, concurrently with forwarding.

Known responses to this problem include (1) to provide greater processing capability, so as to make up for the processor load on the router, (2) to provide only some of these concurrent services, or to provide them only a reduced functionality. While these responses achieve the goal of routing messages in a forwarding network, they have the disadvantage that added services introduce additional load on the router processor and slow down the router.

Accordingly, it would be advantageous to provide a method and system for packet processing that is not subject to drawbacks of the known art.

Summary of the Invention

The invention provides a method and system for packet processing, in which a router (or switch) is capable of processing incoming packets very quickly, thus performing level 2, 3, and 4 routing and switching, and substantial additional services, in

real time. A system includes a packet processing engine (PPE), having elements for receiving packets, distinguishing header and payload information for those packets, outsourcing router decision-making to additional hardware resources (herein a "fast forwarding engine," or FFE), and ultimately forwarding those packets in response to outsourced decisions.

In a first aspect of the invention, the PPE is time-synchronized to the FFE, so that the PPE can send and the FFE can receive packet routing information for decision-making at each one of a sequence of constant-duration time quanta. Similarly, the PPE can receive and the FFE can send packet routing information at each one of a sequence of similar time quanta. In addition to information about where to forward a packet, packet routing information possibly also includes additional information such as packet treatment in response to access control, class of service or quality of service, accounting, and other administrative or managerial criteria.

In a second aspect of the invention, the PPE and the FFE each have separate hardware resources allocated to their functions; these separate hardware resources can include pin count, on chip memory, and transfer bandwidth to off-chip memory. This allows the PPE and the FFE to each perform their functions in parallel without substantial contention for operating resources.

In a third aspect of the invention, multiple PPE and FFE pairs can be incorporated into a scaleable parallel system, operating in parallel to route (or switch) packets among a plurality of input and output interfaces.

In a preferred embodiment, the PPE includes separate treatment of packet header information and payload information, so the amount of information exchanged between the PPE and the FFE, and the amount of actual data movement performed by the PPE, can be relatively minimized. When determining the packet header information, the PPE can also parse the data packet (particularly what is conventionally called the packet header) and extract fields needed by the FFE to perform it's forwarding, ACL and QoS decisions. In this way, the PPE reduces the amount of data that it needs to transmit to the FFE, thereby reducing the number of pins required by both the PPE and the FFE to implement this communication.

Brief Description of the Drawings

Figure 1 shows a block diagram of a system for packet processing and packet forwarding.

Figure 2 shows a process flow diagram of a method of using a packet processing element as in figure 1.

Figure 3 shows a block diagram of a system for parallel packet processing and packet forwarding.

Detailed Description of the Preferred Embodiment

In the following description, a preferred embodiment of the invention is described with regard to preferred process steps and data structures. Those skilled in the art would recognize after perusal of this application that embodiments of the invention can be implemented using circuits adapted to particular process steps and data structures described herein, and that implementation of the process steps and data structures described herein would not require undue experimentation or further invention.

System Elements

Figure 1 shows a block diagram of a system for packet processing and packet forwarding.

A router 100 includes a set of input interfaces 111, a set of output interfaces 112, a packet processing engine (PPE) 120, a PPE memory 130, and a fast forwarding engine (FFE) 140. The router 100 is coupled to one or more communication networks 160.

The router 100 is disposed for routing (or switching) a sequence of packets 170. Each packet 170 includes packet header information 171 and packet payload information 172. Each packet 170 ultimately has packet forwarding information 173 (not shown) decided for it, which is used for routing the packet 170. Each packet 170 might also have a packet index 174 (not shown) for reference purposes.

Packet Processing Engine

The PPE 120 is disposed to perform the following operations:

- The PPE 120 receives input packets 170 at the input interfaces 111.

The input interfaces 111 are coupled to at least one communication network 160.

- The PPE 120 distinguishes packet header information 171 from packet payload information 172.

In a preferred embodiment, input packets 170 and output packets 170 are modified using known packet modification protocols, for which there are known parsing rules. The PPE 120 uses these known parsing rules to distinguish packet header informa-

tion 171 from packet payload information 172. The PPE 120 extracts the packet header information and then stores that packet in the PPE memory 130.

- The PPE 120 records packet header information 171 and packet payload information 172 in the PPE memory 130.

In a preferred embodiment, the PPE 120 uses memory access bandwidth to reference the PPE memory 130 for recording and retrieving packet header information 171 and packet payload information 172 using the PPE memory 130. This allows the PPE 120 to refer to packets by a packet index 174.

- The PPE 120 forwards packet header information 171 to the FFE 140.

In a preferred embodiment, the PPE 120 is ready to forward packet header information 171 to the FFE 140 each two clock cycles. Each clock cycle is preferably 6-7 nanoseconds. It may occur, for any individual incoming packet 170, that the PPE 120 takes much longer than two clock cycles to distinguish packet header information 171 and packet payload information 172. However, the PPE 120 should have at least one new set of packet header information 171 for the FFE 140 at least that often.

Similarly, in a preferred embodiment, the FFE 140 is ready to receive packet header information 171 from the PPE 120 each two clock cycles. It may occur, for

any individual incoming packet 170, that the FFE 140 takes much longer than two clock cycles to decide associated packet forwarding information 173. However, the FFE 140 should be ready to receive one new set of packet header information 171 from the PPE 140 at least that often.

- The PPE 120 receives packet forwarding information 173 for associated packet header information 171 from the FFE 140.

In a preferred embodiment, the PPE 120 uses the packet index 174 to reference both packet header information 171 and associated packet payload information 172 in the PPE memory 130.

- The PPE 120 modifies the packet to generate an output packet 170.

In a preferred embodiment, the PPE 120 performs a rewrite operation on the packet 170. Rewrite operations include adjusting a TTL (time-to-live) IP field, , determining a new CRC, rewriting the MAC-level addresses, and possibly other modifications of the fields.. Rewrite operations, and rewrite rules, are known in the art of Internet packet forwarding.

- The PPE 120 sends output packets 170 from the output interfaces 112.

Similar to the input interfaces 111, the output interfaces 112 are also coupled to at least one communication network 160, preferably the same communication network 160 as the input interfaces 111.

Fast Forwarding Engine

The FFE 140 includes a packet information input port 141, a packet forwarding information output port 142, and is coupled to assistance devices for assisting in making packet forwarding decisions.

The FFE 140 is coupled to a set of routing information memories 143 (including a spanning tree memory and a multicast expansion table), a forwarding content addressable memory (CAM) 144 and a forwarding memory 145, an input access CAM 146 and an output access CAM 147, a CPU 148, and a net-flow routing engine 150.

The FFE 140 is disposed to perform the following operations:

- The FFE 140 receives packet header information 171.

- The FFE 140, with the assistance of the assistance devices, determines packet forwarding information 173 in response to packet header information 171.

In a preferred embodiment, the FFE 140 forwards the packet header information 171 to the forwarding CAM 144, which performs a lookup in its CAM entries to determine packet forwarding information 173 associated with the packet header information 171. Indices responsive to the lookup by the forwarding CAM 144 are recorded in the forwarding memory 145.

The FFE 140 accesses the forwarding CAM 144 to record new forwarding information rules as they become available, such as changes to the perceived network topology, access control, and other administrative or managerial rules. The FFE 140 accesses the forwarding memory 145 to retrieve the packet forwarding information 173 as it is determined.

In a preferred embodiment, the forwarding CAM 144 includes a set of ternary CAM entries. Each ternary CAM entry includes a set of bits which can match to logical 0, to logical 1, or to either (that is, a "don't care" bit). Each ternary CAM entry is thus capable of being matched against the packet header information 171 to determine an index in the forwarding memory 145 of a set of packet forwarding information 173. .

In a preferred embodiment, this additional information is responsive to the IP source address, IP source address, IP source port, IP destination address, IP destination port, protocol type for the packet 170, and whether the packet 170 is unicast or multicast.

In a preferred embodiment, the FFE 140 forwards an identifier for the input interface 111 at which the packet 170 was received to the input access CAM 146, to determine if access is permitted for the packet 170 at that input interface 111.

Similarly, after determining an output interface for the packet 170, the FFE 140 forwards an identifier for the output interface 112 at which the packet 170 was received to the input access CAM 146, to determine if access is permitted for the packet 170 at that output interface 112.

In a preferred embodiment, the packet forwarding information 173 includes how to forward the packet 170 (that is, to which output interface), as well as some or all of the following additional information:

(1) what access control rules (that is, what ACL) to apply to the packet 170;

(2) what class of service (CoS) and quality of service (QoS) rules to apply to the packet 170;

(3) what accounting and statistics to keep regarding the packet 170 or the net flow that the packet 170 is part of;

(4) what other administrative or managerial rules or restrictions to apply to the packet 170.

In a preferred embodiment, this additional information (and other additional services with regard to the packet 170) can be introduced without substantially adding to processing load on the FFE 140, as the forwarding CAM 144 and the forwarding memory 145 provide pattern matching against the packet header information 171.

- The network flow routing engine 150 provides network flow packet forwarding information 173 to the FFE 140, if that network flow packet forwarding information 173 is available.

In a preferred embodiment, if the packet 170 can be routed using network flow information, the network-flow routing engine 150 independently determines networkflow packet forwarding information 173 in response to the network flow associated with the packet header information 171. If the network-flow routing engine 150 is able to determine that network flow packet forwarding information 173, the FFE 140 uses the network flow packet forwarding information 173 in place of packet forwarding information 173 it might otherwise determine for itself.

Method of Operation

Figure 2 shows a block diagram of a packet processing element in a system as in figure 1.

A method 200 includes a set of flow points and a set of steps. The system 100 performs the method 200. Although the method 200 is described serially, the steps of the method 200 can be performed by separate elements in conjunction or in parallel, whether asynchronously, in a pipelined manner, or otherwise. There is no particular requirement that the method 200 be performed in the same order in which this description lists the steps, except where so indicated.

At a flow point 210, the PPE 120 is ready to receive input packets 170 at the input interfaces 111.

At a step 211, the PPE 120 receives an input packet 170 at one of the input interfaces 111.

At a step 212, the PPE 120 parses the packet 170 to distinguish a packet header from a remainder of the packet, and to determine those portions of the packet header that are relevant to packet routing. This allows the PPE 120 to distinguish packet header information 171 from packet payload information 172. The packet 170 is not af-

fected by this parsing. The entire packet 170 remains stored in the PPE memory 130 as one unit.

For example, in a preferred embodiment, the PPE 120 determines the IP source address, IP source port, IP destination address, IP destination port, protocol type for the packet 170, and whether the packet 170 is unicast or multicast. In a preferred embodiment, these values are treated as packet header information 171.

At a step 213, the PPE 120 forwards packet header information 171 for the packet 170 to the FFE 140. As part of this step, the FFE 140 receives packet header information 171 for the packet 170 from the PPE 120.

At a step 214, the FFE 140 sends packet forwarding information 173 for the packet 170 to the PPE 120. As part of this step, the PPE 120 receives packet forwarding information 173 for the packet 170 from the FFE 140.

At a step 215, the PPE 120 associates the packet forwarding information 173 received from the FFE 140 with the packet 170, using the packet index 174.

At a step 216, the PPE 120 rewrites the packet 170 using the packet forwarding information 173 and a set of rewrite rules for the packet 170. As noted above,

rewrite operations include adjusting a hop count for the packet, determining a new CRC, and possibly other protocol reformatting operations.

At a step 217, the PPE 120 sends the packets 170 to the output interface 112 indicated by the packet forwarding information 173.

After a flow point 218, the PPE 120 has sent the packet 170 to a designated output interface 112.

Parallel System

Figure 3 shows a block diagram of a system for parallel packet processing and packet forwarding.

A system 300 for parallel packet processing and packet forwarding includes a plurality of interfaces 110, a plurality of routing pairs 320, and a cross-bar switch 330.

Each plurality of interfaces 110 includes a set of input interfaces 111 and a set of output interfaces 112. Packets 170 can be received at the input interfaces 111 and can be sent using the output interfaces 112.

Each routing pair 320 includes a matched PPE 120 and FFE 140, and associated memories and assistance devices, as described with reference to figure 1.

The cross-bar switch 330 is coupled to outputs from each PPE 120 in each matched routing pair 320.

When a packet 170 is received at a particular interface 110 (and thus at a particular input interface 111 therein), they are coupled to the routing pair 320 associated with that particular interface 110.

When a packet 170 is received at a particular routing pair 320, it is received by the PPE 120 in that particular matched routing pair 320. The PPE 120 and the FFE 140 in that particular routing pair 320 cooperate to route (or switch) and otherwise process the packet 170 as described with regard to figure 1 and figure 2.

When a packet 170 is output from a routing pair 320, the PPE 120 forwards the packet 170 to the crossbar switch 330 with instructions indicating a particular destination interface 110. The crossbar switch 330 provides flow control between different routing pairs 320 so that multiple routing pairs 320 do not simultaneously send packets 170 to the same output interface 112 and overrun buffering therein.

When a packet 170 arrives at the cross-bar switch 330, the cross-bar switch 330 forwards that packet 170 to its destination interface 110, at which it is output from its destination output interface 112.

Alternative Embodiments

Although preferred embodiments are disclosed herein, many variations are possible which remain within the concept, scope, and spirit of the invention, and these variations would become clear to those skilled in the art after perusal of this application.